REMARKS

Claims 1 and 3-35 are currently active.

The Examiner has objected to Claims 10-16. The Examiner has found Claims 17-25 allowable.

Claims 26-35 have been added. Antecedent support for these claims is found on page 10, lines 7-30. Antecedent support for the amendments to Claims 1 and 7 can be found in Claim 5.

The Examiner has rejected Claims 1, 3, 4 and 7-9 as being unpatentable over Palmer. Applicant respectfully traverses this rejection.

Referring to Palmer, there is disclosed a method and apparatus for accessing shared resources with asymmetric safety in a multiprocessing system. Palmer teaches that a problem arises when the communications between nodes 102-104 is interrupted, for example, due to failure of the communication path 108. This problem concerns the nodes competing access to the resource 106, possibly resulting in extremely inefficient operation of the system. In the absence of any scheme for arbitrating disputes between the incommunicant nodes, the

system may experience a thrashing back and forth between the nodes, each node successively fencing the other node from resource access. This situation is undesirable chiefly due to the inefficient time each node spends vying for access to the resource rather than computing or actually accessing the resource. See column 1, lines 50-67. Palmer teaches a multiprocessing system that arbitrates access among multiple competing processing nodes to a shared resource by conducting a membership protocol among all nodes of the system including the shared resource, where the shared resource subsequently fences nodes outside its membership view. See column 2, lines 24-29.

Palmer teaches a membership view is a set of names of application processes participating in a membership group. Palmer teaches the membership view is first invoked and initialized by one of the processes in the system. The processes exchange their local views on the status of the processes in the system. During the view exchange, each process sends it to the other processes its local view on the status of the others. In addition, it receives the views from the other processes, except from those it regards as failed in its local view. See column 7, lines 40-47. The interval of view exchange is terminated if a timeout occurs and each process has not received all the views from those not regarded as failed in its own view. Each process generates a resulting view by intersecting its local view with the set of names of the processes from which it has received the views. See column 7, lines 48-55.

Palmer teaches the system includes active nodes and one passive node. Each active node includes a timer, an invocation counter, and an interval counter. Each active node's timer is set and later expires to affect a timeout condition. Each active node's invocation counter identifies a current instance of membership protocol, as distinguished from earlier or later membership protocols. Each active node's interval counter keeps track of the current interval or round within the presently active membership protocol. See column 11, lines 46-58.

The passive node is the shared resource. The passive node has various sub components, including a processor, storage, timer and a membership area. The membership area includes an invocation counter, an interval counter, multiple membership sub-portions and a passive node view area. See column 11, lines and 9-67.

The passive node's timer and counters have similar functions to the active node's timer and functions. See column 12, lines 27 and 28. The subcomponents of the passive node are used to enable the passive node to participate in a membership protocol, even though the passive node is a passive device relative to the active nodes. The passive node is a passive device in the sense that it serves the active nodes, and may not contain sufficiently powerful computing hardware to participate in a normal membership protocol. See column 12, lines 36-44.

Palmer teaches a membership protocol may be invoked for a number of different reasons. Chiefly, one of the active nodes may invoke a membership protocol when it experiences a communications failure with another node. Another reason, is when a node invokes a membership protocol as a request to join the system. See column 12, lines 60-65. If the executing node is an active node, it exchanges membership views with the active nodes. During the first interval, the active nodes exchange messages with other active nodes to determine their membership views anew, and then exchanging these newly generated views with each other. During subsequent intervals, the active nodes exchange their recent updated views. Next, the executing node performs a membership exchange with the passive node by subscribing to the passive node. The executing node obtains the passive node's membership view by reading the contents of the view area. See column 13, lines 1-18.

The passive node sets its timer to a predetermined value and begins its countdown. This time period is called the membership interval. The active nodes must take certain action, called subscribing, during this period, or else be absent from the passive node's end-of-interval membership view. See column 13, lines 35-40. The passive node notifies the active nodes of the initiation of the current membership interval.

The executing node experiences a timeout condition when its timer expires. If the executing node is an active node, the node notifies the passive node of its timeout. This

ends the current interval of the executing node's participation in the membership protocol. When timeout occurs for the passive node, when the passive node receives no notification of the first active node timeout, the passive node locks the membership area, making it read-only. This is done to actually fix the passive node's membership view as of the end of the membership interval. The membership area reopens as soon as data written to the area cannot affect the membership views. Since the first node to experience an expired timer stops the membership interval despite the other nodes awareness of this fact, this approach is non-blocking. The timeout is guaranteed to occur as long as one active node has access to the shared resource. If one active node in communication with the shared resource fails, that node's timeout cannot end the membership interval, the membership interval will ultimately end, however, when a timeout occurs at another node that has already invoked, or later invokes a membership interval. See column 14, lines 11-30.

Claim 1 has the limitations of "each server includes a state machine and a memory, and maintains in the memory a last time at which each servers' state changed and a value associated with the state when it last changed" and "a disk arbitration mechanism that uses a time stamp-based voting algorithm over the disk blocks associated with the servers and the servers' state to change votes for a primary server". It is respectfully submitted from the description above, that Palmer does not teach or suggest these limitations whatsoever. Palmer does not teach a time stamp based voting algorithm. Applicant specifically chose to use the

limitation "time stamp", and not simply the term "time" in regard to claim 1. What Palmer is teaching is a time based system in regard to when an interval is open or not with respect to a membership. When the interval is open, it just means that certain processes can run, but in no way is there any teaching or suggestion of a disk arbitration mechanism that is time stamp-based.

In contrast, Palmer teaches to use a timer simply for determining whether the membership protocol of each active node is current or not. There is no concern of a servers' state let alone maintaining in the memory a last time at which each servers' state changed and a value associated with the state when it last changed. Palmer teaches each active node includes a timer, an invocation counter, and an interval counter. Each active node's timer is set and later expires to effect a timeout condition,. Each active nodes invocation counter identifies a current instance of membership protocol, as distinguished from earlier or later membership protocols. Each active nodes interval counter keeps track of the current interval or round within the presently active membership protocol. See column 11, lines 47-58. That is the extent to which Palmer teaches to use a timer. Amended Claim 1 of applicant, now has further defined the timestamp-based voting algorithm to include the servers' state, a last time at which each servers' state changed and a value associated with the state when it last changed, to more clearly define applicant's invention of Claim 1 and how it distinguishes over the applied art of record.

Furthermore, this is responsive to the Examiner's comments in the "Response to Arguments" section of the last Office Action, where the Examiner has stated that applicant does not specifically state what the timestamp data that is written comprises and therefore, the Examiner has given a broadest, reasonable interpretation of the claim limitation. By introducing the limitation of state, a last time at which its servers' state changed and a value associated with the state when it last changed into Claim 1, it is in contradistinction to the Examiner's interpretation of Palmer that the passive node will disregard any active nodes attempting to subscribe with an out-of-date invocation counter (column 13, line 51-column 14, line 5 of Palmer). Therefore, it can be seen that because of the timestamp that is recorded by the invocation counter, the passive node can regulate which active nodes may be part of the current membership protocol. See first paragraph of page 3 of the last Office Action. With the added limitations to Claim 1, it is respectfully submitted it is now clear than Palmer does not teach or suggest these newly added limitations and the Examiner's interpretation, is now not applicable.

Furthermore, the Examiner recognizes that Palmer teaches that there is only a single shared storage disk. The Examiner takes the view that it would be obvious to use a plurality of shared storage disks in place of the single disk that Palmer teaches. Applicant respectfully traverses this conclusion. To add a plurality of stores disks in place of the single disk that Palmer teaches would then require the communication network in a protocol to be

established as to how the different storage disks would be utilized in regard to any specific data. It would not be obvious to simply add a plurality of storage disks in such a circumstance and still be able to reliably maintain accurate data over time for the N servers without undue experimentation and development. Accordingly, Claim 1 is patentable over Palmer. Claims 3 and 4 are dependent to parent Claim 1 and are patentable for the reasons Claim 1 is patentable.

Similarly, Claim 7 is patentable for the reasons Claim 1 is patentable. Claims 8 and 9 are dependent to parent Claim 7 and are patentable for the reasons Claim 7 is patentable.

The Examiner has rejected Claims 1, 3, 4 and 7-9 as being unpatentable over Palmer in view of Frank. Applicant respectfully traverses this rejection.

As explained above, Palmer does not teach or suggest the limitation of a "time stamp-based voting algorithm. Frank is totally silent regarding this limitation.

Frank teaches a quorumless cluster using disk-based messaging. Frank teaches a computer network cluster is a collection of interconnected computers which are resources such as data storage. Successful operation of a network cluster requires coordination among the nodes with respect to usage of the shared resources. With multiple users manipulating shared data, precautions must be taken in a network cluster to ensure the data is not corrupted.

A safeguard may be instituted by the network cluster to handle cluster partitioning. Cluster partitioning results when the cluster network degenerates into multiple cluster partitions including a subset of the cluster network nodes, each cluster partition operating independent of each other. These partitions may be the result of one cluster partition having lost network connection with the remaining cluster partitions, the so called partition in spaced problem. To resolve the partition in space problem, the concept referred to as a quorum is typically instituted. This safeguard provides a limited solution to the problem. Requiring a quorum of nodes to be in operation within the network cluster for the network cluster to continue operation, a cluster comprising fewer than a quorum of nodes is forced to terminate operation even though the cluster was operating effectively with the reduced number of nodes. Frank teaches an apparatus and method for implementing a quorumless cluster. See column 1, line 30-column 2, line 36.

Frank teaches a quorumless network cluster 10 having nodes that are connected to shareable storage 22 through a storage connection 24. Frank teaches a single node can form a cluster. The single node can access the shareable storage 22, extract cluster definition data from the storage and form a computer network cluster. The shareable storage may include multiple storage devices. To implement multiple storage devices as shareable storage 22, a header 25 of each storage device may include data indicating the identity of all devices comprising the shareable storage 22. See column 3, lines 8-50.

During operation of the cluster 10, one or more of the member nodes may lose access to disk a. In such a case, it may be decided by the member nodes of the cluster 10 to drop disk a from the cluster. If another node attempted to join the cluster at this time, it could access the header file in disk a which indicates that access to both disk a and disk b is required to gain membership in the cluster 10. If the node did not have access to disk b it would not be able to join the cluster 10.

Frank teaches a cluster manager 32 manages a cluster connected in the computer network cluster 10. The cluster manager 32 can oversee the addition of nodes to and removal of nodes from the computer network cluster 10. It can also prevent the cluster 10 from partition partitioning into multiple cluster partitions. Registration with the cluster manager 32 signifies at the end of the request the changes in cluster membership, among other things, be communicated to the other nodes by the cluster manager 32. A distributed lock manager 34 synchronizes operations between the nodes on the shared resources. See column 4, lines 40-55.

The nodes of the computer cluster 10 are configured in a closed loop arrangement in which each node has a logical previous node and a logical next node. Each node transmits a single heartbeat message to its next node and receives a single heartbeat message from its previous node. This arrangement reduces the number of heartbeat messages

in the cluster to the number of nodes every predetermined time interval. See column 5, lines 40-52. Should any of the nodes fail to receive a heartbeat message from its previous node, it sends a cluster configuration message to each other node in the cluster. In reconfiguration mode, the cluster 10 reverts to an open loop arrangement in which each node sends a heartbeat message into each other node until node membership is once again reconciled. See column 5, line 62-column 6, line 3.

Frank teaches that a cluster definition may be comprised of multiple copies. A joining node to the cluster accesses a map file 52 to determine the location of a current cluster definition. Specifically the adjoining node determines which of two copies of the cluster definition is the current cluster definition. The joining node proceeds to determine a first time stamp for the current definition and read the cluster parameters from the current cluster definition. When the joining node has completed reading the current cluster definition, it again determines the location of the current definition. The second time stamp for the current cluster definition is determined by the node which is compared to the first time stamp. If the two timestamps agree, the joining node reads a valid cluster definition and can now join the cluster 10. If the time stamps do not agree, this indicates that while the joining node was reading the current cluster definition, the coordinator node was in the process of updating the cluster definition. See column 9, lines 40-55. This is the only reference to time or timestamp in Frank. This reference to a time stamp though is in regard to the cluster definition, and has

nothing at all to do with the claim limitation of using a time stamp-based voting algorithm over the disk blocks associated with the servers to change votes for a primary server, as found in Claim 1 of applicant.

Accordingly, neither Palmer nor Frank teach or suggest the limitations of a time stamp-based voting algorithm, a servers' state or "each server includes a state machine and a memory, and maintains in the memory a last time at which each servers' state changed and a value associated with the state when it last changed".

Furthermore, there must be some teaching in the references themselves to apply the teachings the Examiner is relying upon to arrive at applicant's claimed invention, and here, there is none. There is no reason why one skilled in the art would look to a quorumless cluster architecture to combine it with a totally distinct architecture that focuses on membership coordination and is not the least bit concerned with a quorumless cluster architecture.

Moreover, the Examiner cannot take the teachings from each reference out of the context in which it is found. The specific teachings are only applicable to the respective associated context. The use of the plurality of storage devices, as taught by Frank is in the context of a quorumless cluster and can only be used in such an architecture. To remove the

context of this quorumless cluster architecture and simply say that there are a plurality of storage devices which can be applied to the teachings of Palmer and its membership coordination ignores the difficulties of trying to integrate the teachings of Frank into the architecture taught by Palmer. Significant development and design work would be needed to try to figure out how to make such a system operational.

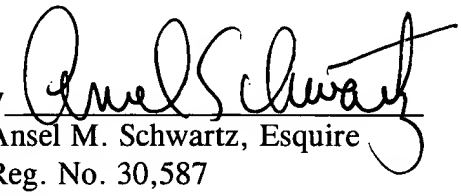
It is respectfully submitted that the Examiner is using hindsight to combine these references of Palmer and Frank. However, hindsight is not patent law. The Examiner cannot use the limitations of applicant's claims as a road map to find the various limitations in disparate references, and having found the different limitations in the different references, conclude that applicant's claimed invention is arrived at.

Accordingly, Claims 1, 3, 4 and 7-9 are patentable over the applied art of record.

In view of the foregoing amendments and remarks, it is respectfully requested that the outstanding rejections and objections to this application be reconsidered and withdrawn, and Claims 1 and 3-35, now in this application be allowed.

Respectfully submitted,

MICHAEL LEON KAZAR

By 

Ansel M. Schwartz, Esquire
Reg. No. 30,587
One Sterling Plaza
201 N. Craig Street
Suite 304
Pittsburgh, PA 15213
(412) 621-9222

Attorney for Applicant